# Supplemental Methods

## Data Set

We used all 9596 cases in TCGA that had germline and tumor DNA, tumor RNA, tumor copy number data and mutation calls in the form of a MAF file. These cases were from 33 disease types comprising 87% of all cases in TCGA. Table 1 shows the breakdown and disease abbreviations.

Table : Cases in the study broken down by disease type. The disease names and abbreviations are from TCGA at https://tcga-data.nci.nih.gov/docs/publications/tcga/

|  |  |  |
| --- | --- | --- |
| **Disease Name** | **Disease Abbreviation** | **Number of Cases** |
| Adrenocortical carcinoma | acc | 77 |
| Bladder urothelial carcinoma | blca | 408 |
| Breast invasive carcinoma | brca | 1034 |
| Cervical squamous cell carcinoma and endocervical adenocarcinoma | cesc | 292 |
| Cholangiocarcinoma | chol | 36 |
| Colon adenocarcinoma | coad | 419 |
| Diffuse large B-cell lymphoma | dlbc | 48 |
| Esophageal carcinoma | esca | 160 |
| Glioblastoma multiforme | gbm | 154 |
| Head and neck squamous cell carcinoma | hnsc | 484 |
| Kidney chromophobe | kich | 65 |
| Kidney renal clear cell carcinoma | kirc | 333 |
| Kidney renal papillary cell carcinoma | kirp | 285 |
| Acute myeloid leukemia | laml | 79 |
| Brain lower grade glioma | lgg | 507 |
| Liver hepatocellular carcinoma | lihc | 367 |
| Lung adenocarcinoma | luad | 510 |
| Lung squamous cell carcinoma | lusc | 493 |
| Mesothelioma | meso | 82 |
| Ovarian serous cystadenocarcinoma | ov | 268 |
| Pancreatic adenocarcinoma | paad | 174 |
| Pheochromocytoma and paraganglioma | pcpg | 178 |
| Prostate adenocarcinoma | prad | 493 |
| Rectum adenocarcinoma | read | 149 |
| Sarcoma | sarc | 252 |
| Skin cutaneous melanoma | skcm | 467 |
| Stomach adenocarcinoma | stad | 373 |
| Testicular germ cell tumors | tgct | 134 |
| Thyroid carcinoma | thca | 489 |
| Thymoma | thym | 118 |
| Uterine corpus endometrial carcinoma | ucec | 533 |
| Uterine carcinosarcoma | ucs | 55 |
| Uveal melanoma | uvm | 80 |

## Mutations

We downloaded MAF files with mutation calls made by MuTect. From these mutations we eliminated ones that:

* Had at least 10 reads in the normal DNA supporting the mutant (based on the n\_alt\_count field in the MAF) or have alt representation in the normal of at least 20% of the rate in the tumor (based on the n\_alt\_count, n\_depth, t\_alt\_count and t\_depth fields in the MAF). This is to remove germline variants incorrectly called as mutants.
* Had no reads covering the tumor DNA, i.e., t\_depth = 0 (which begs the question of how they were called at all).
* Had less than 20% of the tumor DNA supporting the mutation (based on t\_alt\_count and t\_depth), to get rid of mutations in minor subclones.
* Were in intergenic or flanking regions, UTRs or introns or were silent (synonymous) mutations in order to eliminate mutations that were less likely to compromise the function of the gene.
* Were in mitochondrial DNA.

We used the remaining mutations both to classify the mutation count for a particular gene in a given tumor and after additional filtering to explore the expression at mutation sites.

## Germline Variants

We measured allele-specific expression in and around genes by looking at germline heterozygous sites. Because TCGA does not include germline variant calls (only somatic mutations), we variant called all of the normal DNA samples in the data set using FreeBayes.

For our methodology, it was not important to understand exactly what the germline genotype was at all locations. Instead, we just needed to find some very high-confidence heterozygous sites that had RNA expression. Missing a heterozygous site only reduces our ability to see what’s happening, while falsely using a site that’s really not heterozygous could produce erroneous results. Therefore, we erred on the side of caution in selecting the germline variants to use.

We chose those variants that:

* Were single nucleotide variants
* Had an allele frequency of 0.5 (*i.e*., are heterozygous)
* Had an allele balance between 0.4 and 0.6 (were not in subclones or overrepresented for some reason)
* Were not within 150 bases of any other variant called by FreeBayes, regardless of what or how confident the other variant was. This was to make it easier for our software to classify reads as representing alternate or reference alleles
* Had a FreeBayes odds value of more than 20 (which is to say FreeBayes was more than 95% confident it was right); and,
* Have at least 10 reads that cover the variant locus in each of tumor DNA and tumor RNA.

Later in the pipeline we eliminated variants that didn’t have between 40% and 60% of tumor DNA reads representing the alternate allele or that were called as being in a copy number variant in the TCGA copy number files. [Alyssa, please fill in details here.] This was a second way of eliminating sites that either had copy number variants or weren’t in the major clone of the tumor

This resulted in X total variants, which is Y per case or about 1 per every Z bases.

## Tools and Files

Where possible, we did not redo analyses that were already complete in TCGA. In particular, we used their read alignments, mutation calls and copy number variations. Our process required finding germline heterozygous SNVs, which in turn relied on having variant calls for the germline. We used freebayes to generate the VCFs for all of the germline DNA BAM files. The BAMs from TCGA are all aligned to the grch38 human reference genome, and we use that reference exclusively.

Otherwise, we mostly used custom tools that we wrote largely in the C# language. All of our software is open source under the Apache 2.0 license, and is available on Github at <https://github.com/amplab/snap> in the ase branch. We built the software in the SNAP repository because we used some of the code in the SNAP library to, for example, read BAM files. However, we did not use the SNAP aligner itself.

Figure 1 shows the project dataflow, roughly running from top to bottom. This section describes all the boxes in the diagram, again roughly in order of the dataflow.

There is a top-level tool called ASEProcessManager that is not shown in Figure 1 because it is responsible for implementing the dataflow. It inspects the state of the world and generates scripts to do whatever steps are needed to move toward final results.

GenerateMAFConfiguration queries the metadata from TCGA to find the per-disease consolidated MAF files generated by the MuTect tool, and writes their name and file GUIDs into the MAF Configuration file.

GenerateCases reads the TCGA metadata and the MAF Configuration file and finds the list of cases to use in the study. It writes the case GUID, and the GUIDs for all of the files to download into the cases file. It excludes cases that are missing tumor or normal DNA, tumor RNA, copy number or MAF. If a case has tumor or normal methylation, normal copy number or normal RNA it will include those as well.

The cases file is a list of all of the cases, their disease type, the GUIDs and checksums and sizes for their downloadable files, and the pathnames for the downloaded and generated files if they exist. While it is first created by GenerateCases, ASEProcessManager updates the pathnames every time it is run. The other tools use it to find the location of their input files.

ASEProcessManager generates a script to download files that have not yet been downloaded. The dataset contains about 359TB of downloaded files. It also generates a script to compute the MD5 checksum of the downloaded files. Before allowing any subsequent stages to use a downloaded file, it first checks that the compute MD5 sum matches the value from the TCGA metadata and stored in the cases file, in other words that the file downloaded correctly.

Freebayes takes the germline DNA BAM files and generates VCFs for them. While all of the other tools run on Windows, Freebayes is Linux-only. We ran it in Windows Azure virtual machines and downloaded the resulting VCF files by hand. It ran in parallel by using the parallel tool [cite this; it’s quite insistent that you do].

CountReadsCovering takes a BAM file and computes the number of high-quality (MAPQ ≥ 10) reads that map to each locus in the reference genome, as well as the total number of high-quality mapped reads. It writes them into an allcount file. CountReadsCovering is written in C++ and uses the SNAP BAM-reading library routines.

ExtractMAFLines takes the per-disease MAF files, filters the mutation calls and writes them out in per-case files. The filtering removes any mutation call that has at least 10 reads or 20% of all reads in the normal DNA that correspond to the mutation (to get rid of germline variants that were called as somatic mutations). It removes any mutation that has no reads at the mutant locus in the tumor DNA, or that have fewer than 20% of reads in the tumor DNA supporting the mutation (in order to get rid of mutations only in minor subclones). It removes any mutations in mitochondrial DNA and any mutations in intergenic or flanking regions, in UTRs, in introns or that are in exons but that are silent (synonymous) mutations. The intent in selecting mutations like this is that they are likely to result in loss of functional protein (or RNA for non-coding RNAs) in the genes harboring the mutation. This process results in 2,941,789 total selected mutations, of which 53% are missense, 27% are RNA, 9% are splice region, 4% are nonsense, 3% are frame shift deletions, 1% are splice site, 1% are in-frame deletions, 1% are frame shift insertions and < 1% are in-frame insertions, nonstop or translation start site mutations (total not 100% because of round-off error).

SelectGermlineVariants sifts through the germline variant calls made by TCGA in order to find variants used to measure allele-specific expression throughout the genome. The idea is to err on the side of caution. As such, it selects only heterozygous SNVs that have only one non-reference allele, have between 40% and 60% allele balance (a similar test is also applied at later stages), have an odds ratio of more than 20, are at least 150 bases from any other variant called by FreeBayes, and have read depth of at least 10 in tumor DNA and tumor RNA at the mutation site (as determined by the allcount files). From among these variants, it selects at most one per 1000 base region.

GenerateConsolodatedExtractedReads uses samtools to extract the reads from tumor and normal DNA and RNA BAM files near each selected germline variant or mutation. Having these reads separated out from the bulk of the BAM files (and in SAM format rather than compressed BAM) allows other tools easy access to the reads. It is a C++ program that uses the SNAP BAM reading functions.

AnnotateVariants takes the selected variants, extracted MAF lines and extracted reads and determines the count of reads matching the reference or tumor at each mutation/germline variant site for each of tumor and normal DNA and RNA files (including normal RNA in cases where it exists). It produces the annotated selected variants file (which is slightly misnamed, in that it includes both germline variants and somatic mutations).

ExpressionNearMutations takes the annotated variants and measures the mean allele-specific expression in each gene and in each power-of-two times a kilobase region around the gene for each case. It does this both for inclusive regions (i.e., within 64Kb of the gene) and exclusive regions (i.e., between 32Kb and 64Kb). It also does it for the whole autosome, and the whole autosome excluding the chromosome containing the gene.

ExpressionDistribution computes per-disease expression statistics for each genome locus. It computes the fraction and standard deviation of all high quality (MAPQ ≥ 10) mapped RNA bases that map to each location from the tumor sample. By computing in terms of fraction of all mapped bases, it normalizes both for RNA depth-of-coverage and also read length. It’s per-disease because different tumors occur in different cell types, which have different expression profiles.

SelectGenes determines the final set of genes to consider by looking for genes that have at least 30 tumors with at least one mutation in the gene. This results in 8212 genes.

GenerateScatterGraphs takes the annotated somatic mutations from the annotated selected variants file, filters out those that don’t have at least 30 total read depth at the mutation site in both tumor DNA and tumor RNA, adds in expression data for each mutation based on the mapped count of RNA reads and the per-disease expression data for the mutant locus and writes them into per-gene files that contain the data to make the graphs showing tumors on axes of DNA mutant ratio and RNA mutant ratio.

ExpressionByMutationCount takes the allele specific gene expression file and computes how different the distribution of allele-specific expression is based on mutation count (exactly one vs. more than one, and exactly one vs. everything else) for the gene itself and each range around the gene. It uses the Mann-Whitney U test to determine how likely it is that allele specific expression varies. It produces such files both pan-cancer and per-disease

ApplyBonferonniCorrection takes the output of ExpressionByMutationCount and does multiple hypothesis testing correction using the Bonferroni correction. It applies a single correction across all of the genes and ranges (of 375,502 which means that the minimum raw P value that is significant after correction is 2.7x10-8), both per-disease and pan-cancer. While we could have used a less aggressive correction like the Benjamini-Hochberg procedure, with as many tests as we performed it would have made no difference. In addition to the corrected version of the allele specific expression distribution by mutation count file, it also makes a file that lists all the significant results (with significance threshold 0.01).



Figure : Dataflow and the relationship of tools and data.